



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/760,135	01/16/2004	Reiner Hammerich	000005-006700US	7210
68155 7590 11/13/2009				
FOUNTAINHEAD LAW GROUP, PC				
Chad R. Walsh				
900 LAFAYETTE STREET				
SUITE 200				
SANTA CLARA, CA 95050				
EXAMINER				
TSUL, WILSON W				
ART UNIT		PAPER NUMBER		
2178				
NOTIFICATION DATE		DELIVERY MODE		
11/13/2009		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@fountainheadlaw.com
vhernandez@fountainheadlaw.com

Office Action Summary

Application No.

10/760,135

Applicant(s)

HAMMERICH ET AL.

Examiner

WILSON TSUI

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 November 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SE/US)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This final action is in response to the amendment filed on: 06/29/09.
2. Claims 1-24 are pending. Claims 1, 13, and 16 are independent claims.
3. Claims 1-3, 5, 6, 9-13, 15-20, 23, and 24 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri, in view of Bahrs and further in view of Prosise.
4. Claim 4 remains rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri, Bahrs, Prosise, and in further view of Leech.
5. Claim 7 remains rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri, Bahrs, and Prosise, in further view of Lindhorst et al.
6. Claims 8, 14, 17, 21 and 22 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Sheard et al, and Goodwill in further view of Jeyaraman.

Priority

7. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-3, 5, 6, 9-13, 15-20, 23, and 24 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri ("Understanding JavaServer Pages Model 2 architecture", published: December 1999, pages 1-14), in view of Bahrs (US Patent: 6,654,932 B1, issued: Nov. 25, 2003, filed: Oct. 29, 1999) and further in view of Prosisie ("ASP.NET: Web Forms Let You Drag and Drop Your Way to Powerful Apps", pages 1-28).

With regards to claim 1 Seshadri teaches:

Providing a server-side framework to an application (Figure 2: whereas a server-side frame work comprising a model/javabeen, and view/jsp are provided to the application/servlet) *the server side framework being external to the application* (Figure 2: whereas, the application, is represented by the servlet, and the JSP (controller) and Javabeen framework are server side elements that are external to the JSP/controller). *The frame work supporting predefined data types, each data type having a predefined rule* (listing 3, and 4: whereas, the framework supports predefined datatypes, such as string or float datatypes, in a CD object. Each datatype includes an instruction/rule, for the datatype(s) to be initialized with values).

Receiving from an application a request for an object (Listing 3: whereas a getcd function/method is called to request a CD object running on a server), *the request indicating one of the multiple predefined data types* (P9-L7: whereas, the getCD function/method therefore further indicates one of a multiple predefined data types, such as strings or floats for initialization), *the object storing a value of the indicated data type*,

the value being stored in the object in a process format (P9-L7, P9-1: whereas, the setPrice function casts the price data type (in transfer string format), to a process float format), and stored in a CD object as shown in Listing 4).

Creating the object in response to the request (P9-L17: whereas a CD object is created in response to the request).

Generating a markup language page that includes the value in the transfer format read from the object (Listing 2: whereas the values in a CD object are displayed, by inserting the CD attributes in the transfer format/string/ascii into a web page template)

Sending the markup language page to a browser on a client (Figure 4: whereas a markup language page is displayed to a browser on a client).

Receiving a user supplied value in the transfer format from the browser (Listing 3, page 8: whereas a new value is received in the transfer format/string, based on the "ADD" action received).

However, Seshadri does not teach

The object storing a **default** value, ... the **default** value being stored in the object *in a transfer format*, Generating a markup language page that includes *the default value ...*, Storing in the object the user supplied value in the transfer format, The object automatically converting the user supplied value from the transfer format to the process format, the object automatically checking the compliance of the user supplied value in the process format with the predefined rule, and if the user supplied value in the process format complies with the predefined rule, forwarding the user supplied value in

the process format from the object to the application and otherwise automatically resending the markup language page to the browser with the user supplied value in the transfer format.

Yet, even though Sheshadri does not teach the object, Sheshadri still teaches:

- *The value being stored in an object in a transfer format (page 7 and 8 of Sheshadri: whereas, the 'shoppingservlet' class is instantiated into a 'shoppingservlet' object at run-time. The 'shoppingservlet' object storing the value of a token in string format via the 'req' object.)*

Storing in the object the user supplied value in the transfer format (page 7, and 8 of Sheshadri: whereas the 'shoppingservlet' object storing the value of a token in string format via the 'req' object.), The object automatically converting the user supplied value from the transfer format to the process format (Listing 4: whereas, the object automatically stores the float/process format for a price (which was a string in transfer format).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri's CD object, such that the object can store and convert transfer / process format data, as also taught by Sheshadri. The combination would have allowed Sheshadri to have allowed "the processing for all actions carried out within " the [application] (Sheshadri, page 7)).

However, although Sheshadri teaches generating the markup language page using the object which stores a value, Sheshadri does not expressly teach the object stores a **default** value, *the object automatically checking the compliance of the user supplied*

value in the process format with the predefined rule, and if the data complies with the predefined rule, forwarding the user supplied value in the process format from the object to the application and otherwise automatically resending the markup language page to the browser with the user supplied value in the transfer format.

Bahrs et al teaches *the object automatically checking the compliance of the user supplied value in the process format with the predefined rule* (Abstract: whereas a validation object automatically checks the compliance of a new value in a process format with a predefined rule by checking against particular criteria (Fig 87: such as through validation rules)). *And if the user defined data complies with a predefined rule,* (Fig 86: whereas checking includes whether data complies with a predefined rule), *forwarding the user supplied value in a process format to the application* (column 5, lines 8-30, column 16, lines 1-20: whereas, the new value from the user input (such as text data from a text field) is sent/forwarded to an appropriate application) ... *otherwise automatically* generating an exception/alternative-action (Fig 86).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri's object such that the object would have validated input data, as and also such that the object would have further included the ability to forwarded a new value to application, taught by Bahrs et al. The combination of Sheshadri and Bahrs et al would have allowed Sheshadri to have in "response to user input, [made] a call to a validation object" (Bahrs et al, column 3, lines 55-63). However, although the combination of Sheshadri and Bahrs et al teach *otherwise automatically* the combination of Sheshadri and Bahrs et al do not expressly teach

otherwise automatically resending the markup language page to the browser with the user defined value in the transfer format.

Prosise teaches The object storing a **default** value, ... the **default** value being stored in the object *in a transfer format*, Generating a markup language page that includes *the default value ... [and] otherwise automatically resending the markup language page to the browser with the user supplied data in the transfer format, wherein the application processes the data in the processes the data in the process format* (pages 4-7: whereas, default value is null, and using user supplied data is posted back to the user.).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri and Bahrs et al's object and compliance checking methods, such that the object stores a default value; and upon receiving user supplied data, a markup language page with user supplied data, is sent back to the browser, as taught by Prosise. The combination would have allowed Sheshadri, Bahrs et al, and Prosise to have to have "the values entered into text fields don't disappear [upon postback]" (Prosise: page 4).

With regards to claim 2, which depends on claim 1, Sheshadri teaches *wherein the transfer format is a string format*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

With regards to claim 3, which depends on claim 1, the combination of Sheshadri, Bahrs et al, and Prosis, *the predefined rule*, as similarly explained in the rejection for claim 1. Additionally, Bahrs et al teaches *the predefined rule, is internal to the object*, (as explained in the rejection for claim 1), since it is the object that performs the validating, not a separate/external object/entity.

With regards to claim 5, which depends on claim 1, Sheshadri similarly teaches *wherein the operations*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale. However, Sheshadri does not expressly teach the operations *further comprise storing state information in permanent memory and restoring the object by using the state information*.

Yet, Bahrs et al teaches wherein the operations *further comprise storing state information in permanent memory and restoring the object by using the state information* (column 59, lines 25-30: whereas, operations for storing state information is implemented through serialization, and for restoring state information is implemented through deserialization).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri, Bahrs et al, and Prosis's data system, such that state information is stored, and restored for an object, as also taught by Bahrs et al. The combination of Sheshadri, Bahrs et al, and Prosis would have allowed Sheshadri to have "returned a previously created object, or otherwise created a new object, and remembering the object" (Bahrs et al, column 29, lines: 40-54).

With regards to claim 6, which depends on claim 5, the combination of Sheshadri, Bahrs et al, and Prorise teach *wherein the restoring*, as similarly explained in the rejection for claim 5, and is rejected under similar rationale. Additionally, the restoring that is taught by Bahrs et al (as explained in the rejection for claim 5), further includes the restoring *is delayed until transferring* (column 59, lines 25-39: whereas, the restoring is performed until after the object is transferred to the other end).

With regards to claim 9, which depends on claim 1, Sheshadri teaches *wherein the object is provided by the software framework running on a server*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

With regards to claim 10, which depends on claim 1, Sheshadri, Bahrs et al, and Prorise teach *wherein the instructions*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale. Additionally, Bahrs et al further teaches that the data validation system (explained in the rejection for claim 1), further includes the option that the data validation system *do[es] not need to be in a particular programming language* (column 66, lines 50-67: whereas, various types of programming languages can be used to implement the data system of Bahrs et al)

With regards to claim 11, which depends on claim 1, Sheshadri, Bahrs et al, and Prorise teach *wherein the operations*, as similarly explained in the rejection for claim 1,

and is rejected under similar rationale. Additionally, Bahrs et al, further teaches the operations (as explained in the rejection for claim 1) *do not require any particular flow logic* (column 23, lines 40-67: whereas threading support is implemented, such that events/listeners operate in a concurrent manner, without a strict/sequential order)

With regards to claim 12, which depends on claim 1, Sheshadri, Bahrs et al, and Prorise teach *wherein the operations*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale. Additionally Bahrs et al's validation/error-handling scheme, as also explained in the rejection for claim 1, further teaches the validation scheme(s) *do not assume a particular error handling scheme* (whereas, there is no particular single set validation rule, but a plurality of rules that can be set)

With regards to claim 13, for a method performing a similar method as the product in claim 1, is rejected under the same rationale.

With regards to claim 15, which depends on claim 13, for a method performing a similar method as the product in claim 9, is rejected under the same rationale.

With regards to claim 16, for an apparatus performing a similar method as the product in claim 1, is rejected under the same rationale.

With regards to claim 18, which depends on claim 16, for an apparatus performing a similar method as the product in claim 9, is rejected under the same rationale.

With regards to claim 19, which depends on claim 1, the combination of Sheshadri, Bahrs et al, and Prosise teach *wherein the default is a null value*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

With regards to claim 20, which depends on claim 13, the combination of Sheshadri, Bahrs et al, and Prosise teach *wherein the default is a null value*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

With regards to claim 23, which depends on claim 1, the combination of Sheshadri, Bahrs et al, and Prosise teach *applying an input mask to the markup page sent to the browser on the client*, as similarly explained in the rejection for claim 1 (since markup language page with user supplied data (the user supplied data used as a chosen/masked data), is sent back to the browser, as opposed to the input showing a default value), and is rejected under similar rationale.

With regards to claim 24, which depends on claim 23, the combination of Sheshadri, Bahrs et al, and Prosise teach *resending the markup language page to the browser with the user-supplied value in the transfer format includes filling the input*

mask with the user-supplied value in the transfer format, as similarly explained in the rejection for claim 23, and is rejected under similar rationale.

9. Claim 4 remains rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri ("Understanding JavaServer Pages Model 2 architecture", published: December 1999, pages 1-14), Bahrs (US Patent: 6,654,932 B1, issued: Nov. 25, 2003, filed: Oct. 29, 1999), Prosise ("ASP.NET: Web Forms Let You Drag and Drop Your Way to Powerful Apps", pages 1-28), and in further view of Leech (4GuysFromRolla.com, published: December 1, 1999, pages 1-5)

With regards to claim 4, which depends on claim 1, the combination of Sheshadri, Bahrs et al, and Prosise teach *the predefined rule* and *the object*, as explained in the rejection for claim 1, and is rejected under the same rationale. Additionally, Leech teaches the predefined rule (as explained in the rejection for claim 1), further includes wherein *the predefined rule is external to the object* (P2-P3: whereas, the validation rules are enforced at the client side, which is external to the object).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri, Bahrs et al, and Prosise's data system, such that it includes the ability to enforce predefined rules external to the object, as taught by Leech. The combination of Sheshadri, Bahrs et al, Prosise, and Leech would have allowed Sheshadri's system to have further included the ability let "the user know

immediately that something is wrong" (Leech, P3, without having to send the page/form).

10. Claim 7 remains rejected under 35 U.S.C. 103(a) as being unpatentable over Sheshadri ("Understanding JavaServer Pages Model 2 architecture", published: December 1999, pages 1-14), Bahrs (US Patent: 6,654,932 B1, issued: Nov. 25, 2003, filed: Oct. 29, 1999), and Prosis ("ASP.NET: Web Forms Let You Drag and Drop Your Way to Powerful Apps", pages 1-28), in further view of Lindhorst et al (US Patent: 6,981,215 B1, issued: Dec. 27, 2005, filed: Dec. 31, 1998).

With regards to claim 7, which depends on claim 5, Sheshadri, Bahrs et al, and Leech teach *storing state information in permanent memory*, as explained in claim 5, and is rejected under the same rationale. However, the combination of Sheshadri, Bahrs et al, and Leech do not teach the storing of state information in permanent memory *is performed by storing in hidden input fields in the page*

Lindhorst et al teaches the storing of state information in permanent memory *is performed by storing in hidden input fields in the page* (column 14, lines 39-49: whereas, storage/state information is stored in hidden fields in a page).

It would have been obvious to one of the ordinary skill in the art at the time of the invention to have modified Sheshadri, Bahrs et al, and Prosis's system for storing state information to further included the ability to store the state information in hidden input fields in a page as taught by Lindhorst et al. The combination of Sheshadri, Bahrs et al, Prosis, and Lindhorst et al would have allowed Sheshadri to have "simplified the

programmer's task of navigating between pages" (Lindhorst et al, column 6, lines 65-67).

11. Claims 8, 14, 17, 21, and 22 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Sheard et al (US Patent: 6,453,356 B1, issued: Sep. 17, 2002, filed: Apr. 15, 1998), and Goodwill ("Pure Java Server Pages", published: June 08, 2000, Pages: 1-4, 1a, 2a, 3a, 4a, and G1) in further view of Jeyaraman (US Patent: 6,311,187 B1, issued: Oct. 30, 2001, filed: Dec. 29, 1998).

With regards to claim 8, which depends on claim 1, With regards to claim 8, which depends on claim 1, the combination of Sheshadri, Bahrs et al, and Prosis teach *wherein resending the markup language page to the client*, as similarly explained in the rejection for claim 1, and is rejected under similar rationale.

However, Sheshadri, Bahrs et al, and Prosis do not teach Identifying a portion of the markup language page that has changed since the markup language page was previously sent; and resending only the portion of the markup language page that has changed.

Jeyaraman teaches *resending the markup language page to the client includes: identifying a portion of the markup language page that has changed since the markup language page was previously sent* (Abstract: whereas, the identifying includes "determining the differences between the current version of the data at the server and

an older copy of the data at the client"); *and resending only the portion of the markup language page that has changed* (Abstract: whereas, the resending includes "using the differences to construct an update for the copy of the data, which may include node insertion and node deletion operations for hierarchically organized nodes in the data; and sending the update to the client where the update is applied to the copy of the data to produce an updated copy of the data").

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Sheshadri, Bahrs et al, and Prorise's data persistence/state system to further have included the ability to propagate changes since the markup language page has been sent to the client, as taught by Jeyaraman. The combination of Sheshadri, Bahrs et al, Prorise, and Jeyaraman would have allowed Sheshadri's system to have "updated copies of hierarchically structured data" (Jeyaraman, column 1, lines 62-64).

With regards to claim 14, which depends on claim 13, for a method performing a similar method as the product of claim 8, is rejected under the same rationale.

With regards to claim 17, which depends on claim 16, for an apparatus performing a similar method as the product of claim 8, is rejected under the same rationale.

With regards to claim 21, which depends on claim 13, for a method that is similar to the method of claim 8 (whereas since the changed portion is resent, then a write-out/sending action is performed), and is rejected under similar rationale.

With regards to claim 22, which depends on claim 21, the combination of Jeyaraman, Bahrs et al, Prosise, and Jeyaraman teach *the writer function, and a first information stream contains information associated with the portion of the markup language page that has changed*, as similarly explained in the rejection for claim 8, and is rejected under similar rationale.

Furthermore, Jeyaraman's writing function that was disclosed in claim 8, further includes *a second information stream [which] contains information not associated with the portion of the markup language page that as changed, and wherein resending only the portion of the markup language page that has changed includes the second information stream* (Abstract, Fig 5: whereas information for an update includes tree/hierarchical data not associated with the actual value of the portion of the markup language page that has changed, but rather associated with tree operations, when sending a constructed update).

Response to Arguments

12. Applicant's arguments filed 06/29/09 have been fully considered but they are not persuasive.

13. Applicant first argues that the "CD objects are Javabeans that are created in response to requests from the client, not the application as in claim 1, and that as cited from Sheshadri: 'Everytime a user adds an item within Eshop.jsp, the request is posted to the controller servlet. The servlet ... instantiates a new CD bean' [and thus] the request in Sheshadri originates from the user, it does not and cannot indicate a datatype, where the 'object' is used to store a default value having the data type indicated in the request from the application as claimed".

However, this argument is not persuasive since the request for the CD object does come from the application, since the instructions executed/received within the application/controller-servlet are selected for execution to perform the instantiation of a new CD bean/object (Seshadri, page 7). In other words, the CD bean is created in response to the application/controller-servlet requests/instructions for the CD bean to be instantiated. Therefore, the request for creating the object is from the application.

14. The applicant further argues in page 11 of applicant remarks that "unlike the 'object' of claim 1, the CD bean in Sheshadri does not store data in both transfer and process formats, does not convert a user-supplied value from the transfer format to the process format, and does not automatically check the compliance of the user-supplied value in the process format with a predefined rule".

However, the applicant does not expressly give reasons/evidence, as to why the argued limitations are not taught; and just states that the limitations are not taught. Thus the arguments are not persuasive, and the examiner respectfully directs the applicant toward the combination of Sheshadri, Bahrs, and Prosise explained in the rejections above, that explain how combination teaches the argued limitations.

15. The applicant argues in page 11 of applicant remarks that the object of the current application is a server-side construct, which is different in context than Bahrs object, since "Bahrs deals with creating a client development architecture".

However, this argument is not persuasive since the data processing system can be a server (Bahrs, Fig 2, reference 200, bottom of column 38, lines 63-67), and the data processing system performs validation of user input using a validation object (Bahrs, column 3, lines 55-64, column 18, lines 1-4, column 21, lines 18-55: whereas, several implementations of implementing an object with validation on a server).

16. The applicant further argues that Bahrs, col. 15, lines 35-37, "put the examiner's citation in context, specifically states that 'the architectural pattern of the present invention is illustrated as a Java implementation for building thin (or thick) client applications ..' [and thus] validation rules on a client are substantially different than checking compliance of data *from a client* on a server using an "object", as set forth in claim 1".

However, the applicant's argument is not persuasive since the citation about thin or thick client applications is dynamic, meaning the workload/processing can be variable between server and client, depending on client processing capability (Bahrs, column 1, lines 55-57). Thus, Bahrs teachings accommodate for the variability of thin or thick processing, by including several implementations, for which some implementations specifically implement server side objects that perform validation (Bahrs, column 3, lines 55-64, column 18, lines 1-4, column 21, lines 18-55: whereas, several implementations of implementing an object with validation is performed on a server).

17. The applicant argues in page 11 that Prosise does not disclose an "object" or the details set forth in claim 1 [and thus] Prosise is entirely deficient.

However, this argument is not persuasive since the implementation of the server side object has already been explained in the combination of Sheshadri and Bahrs. Furthermore, the Prosise further emphasizes and explains that through object based code, such as a response object (Prosise, page 4), a post back can be implemented, the response object providing default data initially, and then user supplied data is posted back to user (Prosise, bottom of page 4). Thus Prosise supports the use of objects to retain and resend values to the user.

Conclusion

18. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to WILSON TSUI whose telephone number is (571)272-7596. The examiner can normally be reached on Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CESAR B PAULA/
Primary Examiner, Art Unit 2178

/Wilson Tsui/
Patent Examiner
Art Unit: 2178
November 08, 2009